SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO

Sault College

COURSE OUTLINE

COURSE TITLE: INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING

CODE NO. : CSD205   **SEMESTER:** 4

PROGRAM: COMPUTER ENGINEERING TECHNICIAN
COMPUTER PROGRAMMER

AUTHOR: TYCHO BLACK

DATE: Jan. 2001   **PREVIOUS OUTLINE DATED:**   Dec. 1998

APPROVED:

_____   _____
                                     DEAN   **DATE**

TOTAL CREDITS: 5

PREREQUISITE(S): CSD101

HOURS/WEEK: 4

I.   COURSE DESCRIPTION:

This course introduces students to the concepts of Object-Oriented Programming and applies them in practical problem solving exercises. The course presently uses Microsoft Visual C++ as the development environment.

II.   LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:
(Generic Skills Learning Outcomes placement on the course outline will be determined and communicated at a later date.)

A.   Learning Outcomes:

1)   Compare Object-oriented programming (OOP) with procedural programming and compare various OOP languages.

2)   Identify the features of C++ which make it a "better C", and apply them to programming problems.

3)   Demonstrate an understanding of classes, encapsulation and polymorphism by solving programming problems involving their use.

4)   Write programs that use inheritance and exception handling to demonstrate object-oriented design.

B.   Learning Outcomes and Potential Elements of the Performance

Upon successful completion of this course the student will demonstrate the ability to:

1)   Compare Object-oriented programming (OOP) with procedural programming and compare various OOP languages.

Potential Elements of the Performance:

- Identify the most important features of Object-oriented programming languages.
- Assess the strengths and weaknesses of OOP and procedural programming.
- Compare various OOP development environments.
- Compare C++ with Java.

This learning outcome constitutes approximately 5% of the course.

Introduction to Object-oriented Programming   CSD205

2)   Identify the features of C++ which make it a "better C", and apply them to programming problems.

Potential Elements of the Performance:

- Utilize basic stream i/o for input/output.
- Organize program modules into projects and utilize the tools available in an Integrated Development Environment for program development.
- Utilize available methods for passing arguments to functions, including passing by value, passing by reference, passing by pointers and the use of default arguments.
- Utilize enumerator (enum) variables.
- Demonstrate a knowledge of scope and storage classes.
- Use function overloading to provide multiple behaviors for functions.
- Utilize structures (struct) for organizing dissimilar data elements into new data types.
- Use pointers effectively.
- Use file handling methods to be able to create, read, write and update files.
- Use C-type strings and string-handling functions effectively.

This learning outcome constitutes approximately 25% of the course.

3)   Demonstrate an understanding of classes, encapsulation and polymorphism by solving programming problems involving their use.

Potential Elements of the Performance:

- Define classes and implement class members and member functions using appropriate encapsulation (data hiding) mechanisms.
- Declare and define constructors and destructors for classes.
- Use the *this* pointer to point to the invoking object.
- Use dynamic memory allocation and de-allocation with *new* and *delete*.
- Implement various Abstract Data Types (ADT) including a stack and a queue.
- Implement friend functions and friend classes appropriately.

This learning outcome constitutes approximately 45% of the course.

Introduction to Object-oriented Programming   CSD205

4)   Write programs that use overloading, inheritance and exception handling to demonstrate object-oriented design.

Potential Elements of the Performance:

- Implement operator overloading appropriately using friend or member functions.
- Use class inheritance to implement *Is-a* relationships and understand the difference between *composition and inheritance*.
- Use virtual functions (late binding) to redefine class methods for derived classes.
- Implement *Has-a* relationships with member objects (composition) or with private inheritance.
- Use function and class templates to create related functions and classes.
- Use C++ exception handling in programs.
- Create a Win32 program using MFC and Visual C++ (time permitting).

This learning outcome constitutes approximately 25% of the course.

III.   TOPICS:

1)   New features of C++ and elements of C not previously covered.

2)   Overview of the elements and objectives of OOP.

3)   Classes, encapsulation, constructors, destructors, parameter passing.

4)   Friends and operator overloading.

5)   Inheritance, polymorphism and exception handling.

6)   Templates

7)   Win32 programming using the MFC (time permitting)

IV.   REQUIRED RESOURCES/TEXTS/MATERIALS:

Textbook:
"C++ How To Program" 3rd Edition, by Deitel and Deitel, Prentice Hall, 2001

Introduction to Object-oriented Programming   CSD205

V. EVALUATION PROCESS/GRADING SYSTEM

3 WRITTEN TESTS	@ 20% each	60%
ASSIGNMENTS /QUIZZES		40%

(The percentages shown above may vary slightly if circumstances warrant.)

NOTE: *It is required to pass both the theory and the assignment part of this course.   It is not possible to pass the course if a student has a failing average in the three written tests but is passing the assignment portion, (or vice versa).*

TESTS and QUIZZES
Tests will be announced about one week in advance. A zero grade will be given for tests or quizzes missed without a valid reason given in advance. Generally the only valid reasons are medical ones. Quizzes may be unannounced but warning will generally be given.
In some cases assignments will be evaluated with a quiz based on the assigned work, on the due date. Re-writes on these quizzes will not generally be possible so it is essential that assigned work is completed on time and the quizzes written at the required time.

GRADING SYSTEM
A+	90	-	100%
A	80	-	89%
B	70	-	79%
C	60	-	69%
R	Repeat Less than 60%		
X	Incomplete		

PLAGIARISM
While it is expected that students discuss assignments with each other and share ideas, it is not acceptable that students hand in work done by someone else and claim it as their own. Plagiarism on assignments will result in a zero grade being assigned for that assignment for everyone involved.

UPGRADING OF INCOMPLETES
When a student's course work is incomplete or final grade is below 60%, there is the possibility of upgrading to a pass when a student meets all of the following criteria:

Introduction to Object-oriented Programming CSD205

1. The student's attendance has been satisfactory.
2. An overall average of at least 50% has been achieved.
3. The student has not failed all of the theory tests taken.
4. The student has made reasonable efforts to participate in class and complete assignments.

The nature of the upgrading requirements will be determined by the instructor and may involve one or more of the following: completion of existing labs and assignments, completion of additional assignments, re-testing on individual parts of the course or a comprehensive test on the entire course.

ATTENDANCE:
Absenteeism will affect a student's ability to succeed in this course. Absences due to medical or other unavoidable circumstances should be discussed with the instructor. Those whose attendance is poor will not be eligible for an X-grade.

VI. SPECIAL NOTES:

- Special Needs
  If you are a student with special needs (eg. physical limitations, visual impairments, hearing impairments, learning disabilities), you are encouraged to discuss required accommodations with the instructor and/or contact the Special Needs Office, Room E1204, Ext. 493, 717, 491 so that support services can be arranged for you.

- Retention of Course Outlines
  It is the responsibility of the student to retain all course outlines for possible future use in acquiring advanced standing at other post-secondary institutions.

- Course Modifications
  Your instructor reserves the right to make reasonable modifications to the course as deemed necessary to meet the needs of students or take advantage of new or different learning opportunities.

VII. PRIOR LEARNING ASSESSMENT

Students who wish to apply for advanced credit in the course should consult the instructor.